Gradient-Based Solution of Maximum Likelihood Angle Estimation for Virtual Array Measurements

Peter Vouras, Alec Weiss, Maria Becker, Ben Jamroz, Jeanne Quimby, Dylan Williams, Kate Remley
National Institute of Standards and Technology, Gaithersburg, MD

***Abstract*—This paper derives a gradient-based implementation of maximum likelihood angle estimation for virtual planar arrays used to sound millimeter wave channels.**

I. Introduction

Precise measurement and characterization of millimeter wave channels requires antennas capable of high angular resolution to resolve closely spaced multipath sources. To achieve angular resolution on the order of a few degrees these antennas must be electrically large which is impractical for phased array architectures at these frequencies. An alternative approach is to synthesize a virtual aperture in space by using an accurate mechanical positioner to move a receive antenna to points along a sampling grid. An advantage of creating virtual apertures is that the received signal is digitized at every spatial sample position which enables the use of sophisticated angle estimation algorithms such as maximum likelihood techniques. The main contribution of this paper is a new gradient based implementation of maximum likelihood angle estimation that was demonstrated on virtual array data collected at 28 GHz using a vector network analyzer.

II. Overview of Maximum Likelihood Angle Estimation

Consider a virtual array that responds to a single polarization created using N spatial samples taken with an omnidirectional antenna on a planar grid. Assume Q narrowband sources in the far field of the receive antenna impinge on the array from the directions $(\theta_0, \phi_0), (\theta_1, \phi_1), \ldots, (\theta_{Q-1}, \phi_{Q-1})$ in a spherical coordinate system. The complex envelope of the signal received at the ith virtual array element is

$$x_i(t) = \sum_{k=0}^{Q-1} s_k(t) e^{-j\omega_0 \tau_i(\theta_k, \phi_k)} + n_i(t) \tag{1}$$

where $s_k(t)$ denotes the signal emanating from the kth source as observed at the first array element, ω_0 is the center frequency of the sources, $\tau_i(\theta_k, \phi_k)$ is the incremental delay from the first array element to the ith array element for a plane wave impinging on the array from the direction (θ_k, ϕ_k), and $n_i(t)$ is additive zero-mean white measurement noise at the ith element that is assumed to be independent across array elements. Since the sources are narrowband, the interelement delays of the kth signal, $\tau_i(\theta_k, \phi_k)$ for $i = 0, \ldots, N-1$, correspond to incremental phase shifts as the plane wave traverses across the virtual aperture. Thus, (1) can be rewritten as

$$x_i(t) = \sum_{k=0}^{Q-1} s_k(t) e^{-j\frac{2\pi}{\lambda}(md_x u + nd_y v)} + n_i(t) \tag{2}$$

where for $N = N_x N_y,\ m = 0, \ldots, N_x - 1$ is the element index in the x-direction, $n = 0, \ldots, N_y - 1$ is the element index in the y-direction, d_x is the spacing between elements in the x-direction, d_y is the spacing between elements in the y-direction, and $u = \sin\theta\cos\phi$, $v = \sin\theta\sin\phi$.

The array output vector of signals received at each array element can be written as

$$\mathbf{x}(t) = \sum_{k=0}^{Q-1} \mathbf{a}(u_k, v_k) s_k(t) + \mathbf{n}(t) \tag{3}$$

where $\mathbf{x}(t) = [x_0(t), \ldots, x_{N-1}(t)]^T$, $\mathbf{n}(t) = [n_0(t), \ldots, n_{N-1}(t)]^T$ and $\mathbf{a}(u_k, v_k)$ is the $N \times 1$ steering vector of the array in the direction (u_k, v_k) given by,

$$\mathbf{a}(u_k, v_k) = \left[e^{-j\frac{2\pi}{\lambda}(md_x u_k + nd_y v_k)} \Big| 0 \le m \le N_x - 1, 0 \le n \le N_y - 1 \right]^T . \tag{4}$$

Using matrix notation yields,

$$\mathbf{x}(t) = \mathbf{A}(\mathbf{u}, \mathbf{v})\mathbf{s}(t) + \mathbf{n}(t) \tag{5}$$

where $\mathbf{A}(\mathbf{u}, \mathbf{v})$ is the $N \times Q$ matrix of steering vectors

$$\mathbf{A}(\mathbf{u}, \mathbf{v}) = [\mathbf{a}(u_0, v_0), \ldots, \mathbf{a}(u_{Q-1}, v_{Q-1})] \tag{6}$$

and $\mathbf{u} = [u_0, \ldots, u_{Q-1}]^T$, $\mathbf{v} = [v_0, \ldots, v_{Q-1}]^T$, and $\mathbf{s}(t)$ is the $Q \times 1$ vector of signals $\mathbf{s}(t) = [s_1(t) \quad s_2(t), \ldots, s_{Q-1}(t)]^T$. Assume the signals received at each array element are digitized over M time instants $t_0, \ldots, t_{M-1}$. The sampled measurements can be expressed as

$$\mathbf{X} = \mathbf{A}(\mathbf{u}, \mathbf{v})\mathbf{S} + \mathbf{N} \tag{7}$$

with

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}(t_0), \ldots, \mathbf{x}(t_{M-1})]_{N \times M} \\ \mathbf{N} &= [\mathbf{n}(t_0), \ldots, \mathbf{n}(t_{M-1})]_{N \times M} \\ \mathbf{S} &= [\mathbf{s}(t_0), \ldots, \mathbf{s}(t_{M-1})]_{Q \times M} . \end{aligned} \tag{8}$$

The objective of the maximum likelihood (ML) angle estimator is to determine the incoming directions $(u_0, v_0), (u_1, v_1), \ldots, (u_{Q-1}, v_{Q-1})$ of the Q multipath sources from the M array snapshots $\mathbf{x}(t_0), \ldots, \mathbf{x}(t_{M-1})$. The underlying assumptions necessary for deriving the ML estimator are that the number of sources Q is known or can be estimated and that $Q < N$, the steering vectors $\mathbf{a}(u_k, v_k)$ are linearly independent for $k = 0, \ldots, Q-1$, and $E[\mathbf{n}(t)\mathbf{n}(t)^H] = \sigma_n^2 \mathbf{I}$. As described in [1], the computation of the ML angle estimator proceeds by determining the joint probability density function of the sampled data and subsequently the log-likelihood function that must then be maximized with respect to the unknown angles. The final result is that the ML estimates of the directions $(u_0, v_0), (u_1, v_1), \ldots, (u_{Q-1}, v_{Q-1})$ can be obtained by maximizing the function

$$\max_{\mathbf{u},\mathbf{v}} \quad J(\mathbf{u}, \mathbf{v}) = \mathrm{tr}[\mathbf{P}_{\mathbf{A}(\mathbf{u},\mathbf{v})} \widehat{\mathbf{R}}] \tag{9}$$

where the sample covariance matrix is

$$\widehat{\mathbf{R}} = \frac{1}{M} \sum_{k=0}^{M-1} \mathbf{x}(t_k)\mathbf{x}(t_k)^H \tag{10}$$

and $\mathbf{P}_{\mathbf{A}(\mathbf{u},\mathbf{v})}$ is the projection matrix onto the range space of $\mathbf{A}(\mathbf{u}, \mathbf{v})$,

$$\begin{aligned} \mathbf{P}_{\mathbf{A}(\mathbf{u},\mathbf{v})} &= \mathbf{A}(\mathbf{u}, \mathbf{v}) \left[\mathbf{A}(\mathbf{u}, \mathbf{v})^H \mathbf{A}(\mathbf{u}, \mathbf{v})\right]^{-1} \mathbf{A}(\mathbf{u}, \mathbf{v})^H \\ &= \mathbf{A}(\mathbf{u}, \mathbf{v})\mathbf{A}(\mathbf{u}, \mathbf{v})^\dagger, \end{aligned} \tag{11}$$

with † used to denote the pseudoinverse.

A. Alternating Projections Algorithm

The alternating projections (AP) algorithm maximizes the MLE cost function $J(\mathbf{u}, \mathbf{v})$ with respect to one pair of parameters (u_k, v_k) while holding the other parameters fixed. Since iterations of the AP algorithm perform a maximization at every step, the value of $J(\mathbf{u}, \mathbf{v})$ can never decrease, so the algorithm is guaranteed to converge to a local maximum. Depending on the initial conditions, the local maximum may or may not coincide with the global maximum. Since $J(\mathbf{u}, \mathbf{v})$ will in general have many local maxima, proper initialization is vital for the AP algorithm to converge to the global solution.

At the core of the AP algorithm is a projection matrix decomposition described as follows. Consider two arbitrary matrices $\mathbf{X}$ and $\mathbf{Y}$ with the same number of rows. The projection matrix $\mathbf{P}_{[\mathbf{X},\mathbf{Y}]}$ onto the column space of the augmented matrix $[\mathbf{X}, \mathbf{Y}]$ is equal to

$$\mathbf{P}_{[\mathbf{X},\mathbf{Y}]} = \mathbf{P}_{[\mathbf{X},\mathbf{Y}_\mathbf{X}]} \tag{12}$$

where the matrix

$$\mathbf{Y}_\mathbf{X} = \mathbf{P}_\mathbf{X}^\perp \mathbf{Y} = (\mathbf{I} - \mathbf{P}_\mathbf{X})\mathbf{Y}. \tag{13}$$

The symbol $\perp$ is used to denote the orthogonal complement of $\mathbf{P}_\mathbf{X}$. The columns of $\mathbf{Y}_\mathbf{X}$ span the subspace orthogonal to the projection of the range space of $\mathbf{Y}$ onto the range space of $\mathbf{X}$. Since the column space of $\mathbf{Y}_\mathbf{X}$ is orthogonal to the column space of $\mathbf{X}$ and their direct sum spans the column space of $[\mathbf{X}, \mathbf{Y}]$, it follows that

$$\mathbf{P}_{[\mathbf{X},\mathbf{Y}]} = \mathbf{P}_\mathbf{X} + \mathbf{P}_{\mathbf{Y}_\mathbf{X}}. \tag{14}$$

Applying (12) and (14) to $\mathbf{P}_{\mathbf{A}(\mathbf{u},\mathbf{v})}$ yields

$$\begin{aligned} \mathbf{P}_{\mathbf{A}(\mathbf{u},\mathbf{v})} &= \mathbf{P}_{[\mathbf{A}(\widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k), \mathbf{a}(u_k, v_k)]} \\ &= \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k)} + \mathbf{P}_{\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}_k}, \widehat{\mathbf{v}_k})}} \end{aligned} \tag{15}$$

where the $(Q-1) \times 1$ vectors $\widehat{\mathbf{u}}_k$ and $\widehat{\mathbf{v}}_k$ are

$$\begin{aligned} \widehat{\mathbf{u}}_k &= [u_0, u_1, \ldots, u_{k-1}, u_{k+1}, \ldots, u_{Q-1}]^T \\ \widehat{\mathbf{v}}_k &= [v_0, v_1, \ldots, v_{k-1}, v_{k+1}, \ldots, v_{Q-1}]^T \end{aligned} \tag{16}$$

and the $N \times (Q-1)$ matrix $\mathbf{A}(\widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k)$ is

$$\begin{aligned} &\mathbf{A}(\widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k) = \\ &[\mathbf{a}(u_0, v_0), ..., \mathbf{a}(u_{k-1}, v_{k-1}), \mathbf{a}(u_{k+1}, v_{k+1}), ..., \mathbf{a}(u_{Q-1}, v_{Q-1})]\,. \end{aligned} \tag{17}$$

Rewriting the maximization problem in (9) to search along the kth spatial direction (u_k, v_k) at the $(l+1)$st algorithm iteration while holding all other directions fixed yields

$$u_k^{(l+1)}, v_k^{(l+1)} = \arg\max_{u_k, v_k} \quad \mathrm{tr}[\mathbf{P}_{[\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)}), \mathbf{a}(u_k, v_k)]} \widehat{\mathbf{R}}]. \tag{18}$$

Equation (18) states that to obtain the angle estimates $u_k^{(l+1)}, v_k^{(l+1)}$ for the kth source at the $(l+1)$st algorithm iteration, the parameters $\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)}$ are held fixed while the parameters u_k, v_k are free to vary. Applying the matrix decomposition in (14) to (18) and ignoring the first term in the summation since it is constant yields the equivalent maximization problem

$$u_k^{(l+1)}, v_k^{(l+1)} = \arg\max_{u_k, v_k} \quad \mathrm{tr}[\mathbf{P}_{\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}} \widehat{\mathbf{R}}]. \tag{19}$$

Using (13) and (11) the vector $\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}$ can be written as

$$\begin{aligned} \mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})} &= \left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right] \mathbf{a}(u_k, v_k) \\ &= \left[\mathbf{I} - \mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)}) \mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})^\dagger\right] \mathbf{a}(u_k, v_k). \end{aligned} \tag{20}$$

Equation (20) shows that the vector $\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}$ is orthogonal to the projection of $\mathbf{a}(u_k, v_k)$ onto the column space of $\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})$. Also by (11)

$$\begin{aligned} &\mathbf{P}_{\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}} = \\ &= \frac{\left[\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right] \left[\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right]^H}{\left[\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right]^H \left[\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right]}. \end{aligned} \tag{21}$$

Define the unit norm vector

$$\mathbf{b}_k^{(l)} \equiv \mathbf{b}(u_k, v_k; \widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)}) = \frac{\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}}{\|\mathbf{a}(u_k, v_k)_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\|_2} \tag{22}$$

and substitute (21) into (20). By applying properties of the trace operator including $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$, the optimization problem in (19) becomes

$$\begin{aligned} u_k^{(l+1)}, v_k^{(l+1)} &= \arg\max_{u_k, v_k} \quad \mathbf{b}_k^{(l)^H} \widehat{\mathbf{R}} \mathbf{b}_k^{(l)} \\ &\equiv \arg\max_{u_k, v_k} \quad J^l(u_k, v_k). \end{aligned} \tag{23}$$

The entire AP algorithm can now be summarized as follows,

Algorithm 1. MLE-AP Algorithm to Compute Angles of Arrival

Require: Initial or a priori angle estimates $u_0^{(0)}, \dots, u_{Q-1}^{(0)}$ and $v_0^{(0)}, \dots, v_{Q-1}^{(0)}$

1: Set algorithm iteration $l = 1$

2: Until $\left|u_k^{(l+1)} - u_k^{(l)}\right|^2 < \epsilon$ and $\left|v_k^{(l+1)} - v_k^{(l)}\right|^2 < \epsilon$ for all $k = 0, \dots, Q-1$, compute the AOA estimates for the kth source by solving $u_k^{(l+1)}, v_k^{(l+1)} = \arg\max_{u_k, v_k} \quad J^l(u_k, v_k)$

The two primary contributions of this paper described in the next sections are a successful approach to initialize the MLE-AP algorithm and a gradient-based method to maximize the cost function $J^l(u_k, v_k)$ at each iteration.

III. Gradient-Based Implementation of MLE-AP Algorithm

A. Derivation of Gradient Vector

In this section an analytical expression for the gradient vector of the $(l+1)$st cost function specified in (23) is derived. To start, rewrite the cost function as

$$J^l(u_k, v_k) = \tag{24}$$

$$\frac{\mathbf{a}(u_k, v_k)^H \left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right]^H \widehat{\mathbf{R}} \left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right] \mathbf{a}(u_k, v_k)}{\mathbf{a}(u_k, v_k)^H \left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right] \mathbf{a}(u_k, v_k)}$$

$$\equiv \frac{\mathbf{a}(u_k, v_k)^H \mathbf{B} \mathbf{a}(u_k, v_k)}{\mathbf{a}(u_k, v_k)^H \mathbf{Q} \mathbf{a}(u_k, v_k)}$$

by substituting (20) into (23) and noting that the idempotent and self-adjoint properties of orthogonal projection matrices imply that $\left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right]^H \left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right] = \left[\mathbf{I} - \mathbf{P}_{\mathbf{A}(\widehat{\mathbf{u}}_k^{(l)}, \widehat{\mathbf{v}}_k^{(l)})}\right]$.

Note that the steering vector $\mathbf{a}(u_k, v_k)$ defined in (4) can be rewritten as the Kronecker product of two steering vectors,

$$\mathbf{a}(u_k, v_k) = \left[e^{-j\frac{2\pi}{\lambda} m d_x u_k}\right]^T \otimes \left[e^{-j\frac{2\pi}{\lambda} n d_y v_k}\right]^T \tag{25}$$

with $0 \le m \le N_x - 1$ and $0 \le n \le N_y - 1$. Hereafter, to simplify notation, the subscript k denoting the kth AOA source will be dropped from the coordinates u, v. Next consider a change in pointing direction corresponding to (δ_u, δ_v). Then,

$$\begin{aligned} &\mathbf{a}(u + \delta_u, v + \delta_v) = \\ &\left[e^{-j\frac{2\pi}{\lambda} m d_x (u + \delta_u)} \middle| 0 \le m \le N_x - 1\right]^T \otimes \\ &\left[e^{-j\frac{2\pi}{\lambda} n d_y (v + \delta_v)} \middle| 0 \le n \le N_y - 1\right]^T. \end{aligned} \tag{26}$$

Define the diagonal matrices,

$$\begin{aligned} \mathbf{\Delta} u &= \delta_u \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2\pi}{\lambda} d_x & 0 & 0 & 0 \\ 0 & 0 & \frac{2\pi}{\lambda} 2 d_x & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \frac{2\pi}{\lambda}(N_x - 1) d_x \end{bmatrix} \\ &= \delta_u \mathbf{T}_x \end{aligned} \tag{27}$$

and similarly $\mathbf{\Delta} v = \delta_v \mathbf{T}_y$ using d_y instead of d_x. Then using the identity

$$\mathbf{\Delta} u \oplus \mathbf{\Delta} v = \mathbf{\Delta} u \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{\Delta} v \tag{28}$$

and the properties of the matrix exponential, the perturbed steering vector $\mathbf{a}(u + \delta_u, v + \delta v)$ can be written as

$$\begin{aligned} \mathbf{a}(u + \delta_u, v + \delta v) &= e^{-j\mathbf{\Delta} u} \otimes e^{-j\mathbf{\Delta} v} \mathbf{a}(u, v) \\ &= e^{-j(\mathbf{\Delta} u \oplus \mathbf{\Delta} v)} \mathbf{a}(u, v). \end{aligned} \tag{29}$$

At this point it is useful to clarify the overarching strategy for computing the gradient vector of the cost function $J(u, v)$ in (24), where the superscript iteration index l has been dropped for simplicity. A related approach is also described in [2]. The desired gradient vector of $J(u, v)$ to be computed is defined as $\nabla J = \left[{\partial J}/{\partial u} \quad {\partial J}/{\partial v}\right]^T$. In terms of numerator and denominator functions, $J(u, v) = {N(u,v)}/{D(u,v)}$, so using the quotient rule for differentiation yields

$$\begin{aligned} \frac{\partial J}{\partial u} &= \frac{\frac{\partial N}{\partial u} D(u, v) - \frac{\partial D}{\partial u} N(u, v)}{D(u, v)^2} \\ \frac{\partial J}{\partial v} &= \frac{\frac{\partial N}{\partial v} D(u, v) - \frac{\partial D}{\partial v} N(u, v)}{D(u, v)^2}. \end{aligned} \tag{30}$$

It is clear that to apply the quotient rule for computing ∇J it is also necessary to compute $\nabla N = \left[{\partial N}/{\partial u} \quad {\partial N}/{\partial v}\right]^T$ and $\nabla D = \left[{\partial D}/{\partial u} \quad {\partial D}/{\partial v}\right]^T$. A useful fact is that since the numerator function N is continuously differentiable with respect to u and v, the directional derivative $N'(\mathbf{p}; \mathbf{d})$ of N at the point $\mathbf{p} = [u \quad v]^T$ in the direction $\mathbf{d} = [\delta_u \quad \delta_v]^T$ is equal to [3]

$$N'(\mathbf{p}; \mathbf{d}) = \nabla N(\mathbf{p})^T \mathbf{d}. \tag{31}$$

In the case at hand, ∇N is unknown and the quantity to be determined, but the directional derivative $N'(\mathbf{p}; \mathbf{d})$ can also be calculated as the derivative with respect to t of the function $G_N(t) = N(\mathbf{p} + t\mathbf{d})$ evaluated at $t = 0$,

$$N'(\mathbf{p}; \mathbf{d}) = \left.\frac{d}{dt} G_N(t)\right|_{t=0} = \left.\frac{d}{dt} N(\mathbf{p} + t\mathbf{d})\right|_{t=0}. \tag{32}$$

Thus ∇N can be recovered by using (32) to compute the directional derivative $N'(\mathbf{p}; \mathbf{d})$ and then writing the result in a form compatible with (31) to recover the gradient vector. The same procedure also applies to the denominator function $D(u, v)$ using the derivative with respect to t of the function $G_D(t) = D(\mathbf{p} + t\mathbf{d})$ evaluated at $t = 0$.

Continuing along this tack and starting with $D(u, v)$ yields,

$$\begin{aligned} G_D(t) &= D(u + t\delta_u, v + t\delta_v) = \\ &= \mathbf{a}(u + t\delta_u, v + t\delta_v)^H \mathbf{Q} \mathbf{a}(u + t\delta_u, v + t\delta_v) \\ &= \mathbf{a}(u, v)^H e^{j(\mathbf{\Delta} u \oplus \mathbf{\Delta} v)t} \mathbf{Q} e^{-j(\mathbf{\Delta} u \oplus \mathbf{\Delta} v)t} \mathbf{a}(u, v) \end{aligned} \tag{33}$$

and the desired directional derivative

$$
\begin{aligned}
D'(\mathbf{p};\mathbf{d}) &= j\mathbf{a}(u,v)^H\left[\boldsymbol{\Delta}u\oplus\boldsymbol{\Delta}v,\mathbf{Q}\right]\mathbf{a}(u,v) \quad (34)\\
&= j\mathrm{tr}\left(\left[\boldsymbol{\Delta}u\oplus\boldsymbol{\Delta}v,\mathbf{Q}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\\
&= j\mathrm{tr}\left(\left[\delta_v\mathbf{T}_y\oplus\delta_u\mathbf{T}_x,\mathbf{Q}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\\
&= j\mathrm{tr}\left(\left(\left[\delta_u\mathbf{T}_x\otimes\mathbf{I},\mathbf{Q}\right]+\left[\mathbf{I}\otimes\delta_v\mathbf{T}_y,\mathbf{Q}\right]\right)\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\\
&= j\left[\mathrm{tr}\left(\left[\delta_u\mathbf{T}_x\otimes\mathbf{I},\mathbf{Q}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\right.\\
&\qquad\left.+\mathrm{tr}\left(\left[\mathbf{I}\otimes\delta_v\mathbf{T}_y,\mathbf{Q}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\right]\\
&= j\left[\delta_u\mathrm{tr}\left(\left[\mathbf{T}_x\otimes\mathbf{I},\mathbf{Q}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\right.\\
&\qquad\left.+\delta_v\mathrm{tr}\left(\left[\mathbf{I}\otimes\mathbf{T}_y,\mathbf{Q}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\right]
\end{aligned}
$$

where the notation $[\mathbf{A},\mathbf{B}]$ denotes the Lie bracket, $[\mathbf{A},\mathbf{B}]=\mathbf{AB}-\mathbf{BA}$. Rewriting (34) in matrix form and comparing to (31) yields

$$
D'(\mathbf{p};\mathbf{d}) = \quad (35)
$$

$$
\begin{bmatrix} -\mathrm{imag}\left(\mathrm{tr}\left(\left[\mathbf{T}_x\otimes\mathbf{I},\mathbf{Q}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\right)\\ -\mathrm{imag}\left(\mathrm{tr}\left(\left[\mathbf{I}\otimes\mathbf{T}_y,\mathbf{Q}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\right)\end{bmatrix}^T\begin{bmatrix}\delta_u\\ \delta_v\end{bmatrix}
$$

$$
\equiv \nabla D(\mathbf{p})^T\mathbf{d}. \quad (36)
$$

Repeating the same argument for the numerator function $N(u,v)$ results in

$$
N'(\mathbf{p};\mathbf{d}) = \quad (37)
$$

$$
\begin{bmatrix} -\mathrm{imag}\left(\mathrm{tr}\left(\left[\mathbf{T}_x\otimes\mathbf{I},\mathbf{B}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\right)\\ -\mathrm{imag}\left(\mathrm{tr}\left(\left[\mathbf{I}\otimes\mathbf{T}_y,\mathbf{B}\right]\mathbf{a}(u,v)\mathbf{a}(u,v)^H\right)\right)\end{bmatrix}^T\begin{bmatrix}\delta_u\\ \delta_v\end{bmatrix}
$$

$$
\equiv \nabla N(\mathbf{p})^T\mathbf{d}. \quad (38)
$$

Now the components of ∇N and ∇D are clearly available to substitute into (30) to compute ∇J.

B. Conjugate Gradient Algorithm

The conjugate gradient algorithm for maximizing the cost function in (23) for the kth AOA source at the lth iteration of the MLE-AP algorithm is,

Algorithm 2. Conjugate Gradient Algorithm

Require: Initial angle estimates u_0 and v_0. One approach for obtaining u_0 and v_0 is to compute (39) over all possible angles and choose the peaks in the output.

1: Set the initial search direction $\mathbf{d}_0 = \nabla J(u_0, v_0)$
2: Until $\|\nabla J(u_j, v_j)\|_2 \leq \epsilon$, where j denotes the conjugate gradient iteration index, do the following:
3: Determine the step-size μ_j
4: Set $\mathbf{p}_{j+1} = \mathbf{p}_j + \mu_j \mathbf{d}_j$ where $\mathbf{p}_j = [u_j \quad v_j]^T$
5: Set $\mathbf{g}_{j+1} = \nabla J(u_{j+1}, v_{j+1})$
6: Set $\mathbf{d}_{j+1} = \mathbf{g}_{j+1} + \alpha_j \mathbf{d}_j$
7: Set $\alpha_j = \frac{\mathbf{g}_{j+1}^T(\mathbf{g}_{j+1}-\mathbf{g}_j)}{\mathbf{g}_j^T\mathbf{g}_j}$
8: Set $j = j + 1$

The step-size μ_j for the jth conjugate gradient iteration can be set equal to a constant value small enough to ensure algorithm convergence or it can be chosen via a one-dimensional line search. The preferred approach is to use Armijo's rule. Given an initial stepsize, $0 < \rho < 1$, Armijo's rule chooses the final stepsize μ_j to be the first value in the sequence $1, \rho, \rho^2, \rho^3, \ldots$ that satisfies the condition $J(\mathbf{p}_j + \mu_j \mathbf{d}_j) \geq J(\mathbf{p}_j) + \mu_j \alpha \nabla J(\mathbf{p}_j)^T \mathbf{d}_j$, for a fixed scalar $0 < \alpha < 0.5$. In other words, $\mu_j = \rho^m$ for some integer m.

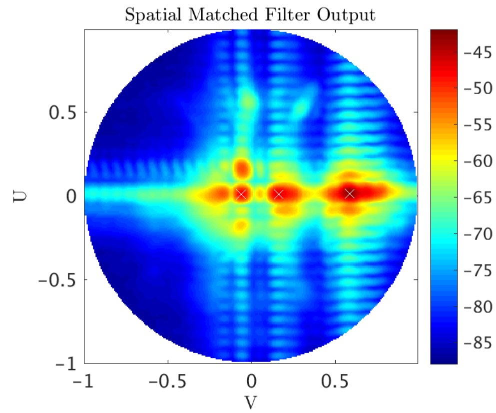

Fig. 1. Output of Spatial Matched Filter

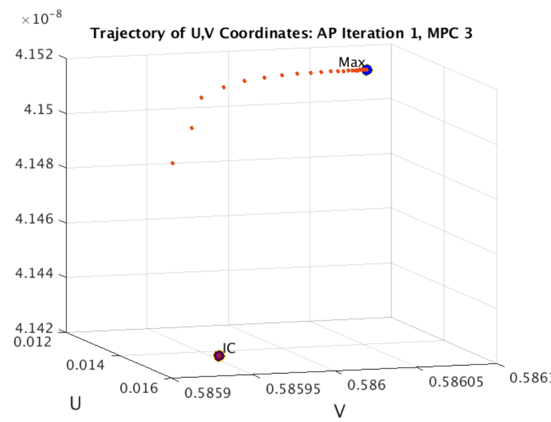

Fig. 2. Algorithm Convergence Trajectory: Multipath Cluster 3

IV. Measured Results and Conclusions

Since the cost function $J(\mathbf{p})$ has multiple peaks it is necessary to properly initialize the conjugate gradient algorithm to ensure that it converges to the correct solution. A simple approach is to compute the output of a spatial matched filter s for all u, v directions in space as given by

$$
s(u,v) = \mathbf{a}(u,v)^H\widehat{\mathbf{R}}\mathbf{a}(u,v) \quad (39)
$$

and then choose the largest peaks for initial angle estimates. Figure 1 illustrates the output of the spatial matched filter for virtual array data collected in a lab room. The white crosses correspond to the multipath peaks detected using a thresholding algorithm. These peaks are used to initialize the conjugate gradient MLE algorithm. Figure 2 illustrates the algorithm's path to convergence for the third multipath source. In all cases the algorithm converged within less than 40 iterations.

References

[1] I. Ziskind and M. Wax, "Maximum Likelihood Localization of Multiple Sources by Alternating Projection," *IEEE Transactions on Acoustics,*

Speech, and Signal Processing, Vol. 36, No. 10, October 1988, pp. 1553-1560.
[2] S. T. Smith, "Optimum Phase-Only Adaptive Nulling," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 47, No. 7, July 1999, pp. 1835-1843.
[3] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1995.